

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開2001-251525

(P2001-251525A)

(43)公開日 平成13年9月14日(2001.9.14)

(51) Int.Cl. <sup>7</sup>		識別番号	F I	アコード <sup>7</sup> (参考)
H 0 4 N	1/60		C 0 6 F 3/12	L 5 B 0 1 4
G 0 6 F	3/12		9/06	4 1 0 Q 5 B 0 2 1
	9/06	4 1 0	13/10	3 2 0 A 5 B 0 7 6
	13/10	3 2 0	H 0 4 N 1/40	D 5 C 0 7 7
H 0 4 N	1/46		1/46	Z 5 C 0 7 9
審査請求 未請求 請求項の数30 O L (全 12 頁)				

(21) 出願番号 特願2000-58673(P2000-58673)

(22) 出願日 平成12年3月3日(2000.3.3)

(71)出題人 000001007

キヤノン株式会社

東京都大田区下丸子3丁目30番2号

(72) 究明者 伊藤 康治

東京都大田

ン株式会社内

100090538

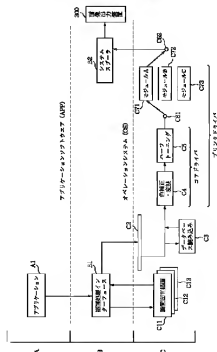
最終頁に続く

(54)【発明の名称】 情報処理装置、情報処理方法及びデバイスドライバプログラムを格納した記憶媒体

(57) 【要約】

【課題】 複数のデバイスに共通するコアドライバモジュールとデバイス特有な処理に依存する依存モジュールとから構成されるデバイスドライバにおいて、既存のモジュールと同様の処理を行う、新規デバイスの追加作業の際に発生する依存モジュール内部の検査部分の書き換え作業をなくすことを目的とする。

【解決手段】 画像出力装置のデバイス ID に応じて使用する依存モジュールを切り替える情報処理方法において、入力されるデバイス ID に基づいてデータベースを検索することにより依存モジュールに対応するプロセス ID を取得し、このプロセス ID を依存モジュールに引き渡すことによりデバイスと依存モジュールとの対応に関するエラーチェックを行わせる。



## 【特許請求の範囲】

【請求項1】 画像出力装置に送信すべき画像情報を生成する情報処理装置であって、

入力される描画情報に対して、前記画像出力装置のタイプに共通の処理を行う共通処理手段と、

前記画像出力装置のタイプに依存する処理を行い、前記画像情報を生成する複数の依存処理手段と、

それぞれの画像出力装置に対して、使用する依存処理手段を識別する依存処理手段識別情報と前記依存処理手段に渡すべき処理識別情報とを保持するデータベース管理手段とを有し、

前記共通処理手段は、前記依存処理手段識別情報に基づいて使用する依存処理手段を選択し、選択された依存処理手段に対して前記処理識別情報を渡すことにより、前記依存処理手段による内部検査を行わせるよう制御することを特徴とする情報処理装置。

【請求項2】 前記データベース管理手段は、更に、前記共通処理手段で使用する色補正データを保持していることを特徴とする請求項1記載の情報処理装置。

【請求項3】 前記共通処理手段は、オペレーティングシステムから受け取るディスプレイ表色系データを、前記色補正データに基づいて、画像出力装置における表色系データに色変換処理することを特徴とする請求項2記載の情報処理装置。

【請求項4】 前記依存処理手段は、オペレーティングシステムから受け取るRGBラスタデータを、前記色補正データに基づいて、CMYKもしくはCMYK淡C淡Mのラスタデータに色変換処理することを特徴とする請求項3記載の情報処理装置。

【請求項5】 前記共通処理手段は、画像出力装置に対応するデバイス識別情報に基づいて、前記データベース管理手段から前記依存処理手段識別情報と前記処理識別情報とを取得することを特徴とする請求項1乃至4のいずれかに記載の情報処理装置。

【請求項6】 前記デバイス識別情報を双方向インタフェースを介して前記画像出力装置から取得するデバイス識別情報取得手段を更に有することを特徴とする請求項5記載の情報処理装置。

【請求項7】 情報処理装置に新規に画像出力装置による出力を制御するためのデバイスドライバをインストールするインストール手段を更に有し、

前記インストール手段は、前記データベース管理手段により新規にインストールする画像出力装置のデバイス識別情報に対応する前記依存処理手段がすでに組み込まれている場合は、前記依存処理手段を新たに組み込まないことを特徴とする請求項1乃至6のいずれかに記載の情報処理装置。

【請求項8】 前記インストール手段は、前記データベース管理手段により新規にインストールする画像出力装置のデバイス識別情報に対応する前記依存処理手段がすでに組み込まれている場合は、新規の画像出力装置に対応するアイコンを作成することを特徴とする請求項7記載の情報処理装置。

【請求項9】 前記インストール手段は、前記データベース管理手段のバージョン情報を判断し、新規にインストールしようとするバージョン情報の方が新しい場合は、データベース管理手段を上書きすることを特徴とする請求項7乃至8のいずれかに記載の情報処理装置。

【請求項10】 前記処理識別情報は、前記依存処理手段に対応するすべてのデバイス識別情報の中で最も古いデバイス識別情報と同じ値であることを特徴とする請求項5記載の情報処理装置。

【請求項11】 画像出力装置に送信すべき画像情報を生成する情報処理方法であって、

入力される描画情報に対して、前記画像出力装置のタイプに共通の処理を行う共通処理工程と、

前記画像出力装置のタイプに依存する処理を行い、前記画像情報を生成する複数の依存処理工程と、

それぞれの画像出力装置に対して、使用する依存処理工程を識別する依存処理工程識別情報と前記依存処理工程に渡すべき処理識別情報とをデータベースで管理する管理工程とを有し、

前記共通処理工程は、前記依存処理工程識別情報に基づいて使用する依存処理工程を選択し、選択された依存処理工程に対して前記処理識別情報を渡すことにより、前記依存処理工程による内部検査を行わせるよう制御することを特徴とする情報処理方法。

【請求項12】 前記データベースは、更に、前記共通処理工程で使用する色補正データを保持していることを特徴とする請求項11記載の情報処理方法。

【請求項13】 前記共通処理工程は、オペレーティングシステムから受け取るディスプレイ表色系データを、前記色補正データに基づいて、画像出力装置における表色系データに色変換処理することを特徴とする請求項12記載の情報処理方法。

【請求項14】 前記共通処理工程は、オペレーティングシステムから受け取るRGBラスタデータを、前記色補正データに基づいて、CMYKもしくはCMYK淡C淡Mのラスタデータに色変換処理することを特徴とする請求項13記載の情報処理方法。

【請求項15】 前記共通処理工程は、画像出力装置に対応するデバイス識別情報に基づいて、前記データベースから前記依存処理工程識別情報と前記処理識別情報とを取得することを特徴とする請求項11乃至14のいずれかに記載の情報処理方法。

【請求項16】 前記デバイス識別情報を双方向インタフェースを介して前記画像出力装置から取得するデバイス識別情報取得工程を更に含むことを特徴とする請求項15記載の情報処理方法。

【請求項17】 情報処理装置に新規に画像出力装置に

よる出力を制御するためのデバイスドライバをインストールするインストール工程を更に含み、

前記インストール工程は、新規にインストールする画像出力装置のデバイス識別情報に対応する前記依存処理工程がすでに組み込まれている場合は、前記依存処理工程を新たに組み込まないことを特徴とする請求項1乃至16のいずれかに記載の情報処理方法。

【請求項18】 前記インストール工程は、新規にインストールする画像出力装置のデバイス識別情報に対応する前記依存処理工程がすでに組み込まれている場合は、新規の画像出力装置に対応するアイコンを作成することを特徴とする請求項17記載の情報処理方法。

【請求項19】 前記インストール工程は、前記データベースのバージョン情報を判断し、新規にインストールしようとするバージョン情報の方が新しい場合は、データベースを上書きすることを特徴とする請求項17乃至18のいずれかに記載の情報処理方法。

【請求項20】 前記処理識別情報は、前記依存処理工程に対応するすべてのデバイス識別情報の中で最も古いデバイス識別情報と同じ値であることを特徴とする請求項15記載の情報処理方法。

【請求項21】 画像出力装置に送信すべき画像情報を生成するデバイスドライバプログラムを格納した記憶媒体であって、

入力される描画情報に対して、前記画像出力装置のタイプに共通の処理を行う共通処理モジュールと、

前記画像出力装置のタイプに依存する処理を行い、前記画像情報を生成する複数の依存処理モジュールと、

それぞれの画像出力装置に対して、使用する依存処理モジュールを識別する依存処理モジュール識別情報と前記依存処理モジュールに渡すべきプロセス識別情報とを管理するデータベースとを供給し、

前記共通処理モジュールは、前記依存処理モジュール識別情報に基づいて使用する依存処理モジュールを選択し、選択された依存処理モジュールに対して前記プロセス識別情報を渡すことにより、前記依存処理モジュールによる内部検査を行わせるよう制御することを特徴とするコンピュータ読み取り可能なデバイスドライバプログラムが格納された記憶媒体。

【請求項22】 前記データベースは、更に、前記共通処理モジュールで使用する色補正データを保持していることを特徴とする請求項21記載の記憶媒体。

【請求項23】 前記共通処理モジュールは、オペレーティングシステムから受け取るディスプレイ表色系データを、前記色補正データに基づいて、画像出力装置における表色系データに変換処理することを特徴とする請求項22記載の記憶媒体。

【請求項24】 前記共通処理モジュールは、オペレーティングシステムから受け取るRGBラスターデータを、前記色補正データに基づいて、CMYKもしくはCMY

K淡C淡Mのラスターデータに変換処理することとを特徴とする請求項23記載の記憶媒体。

【請求項25】 前記共通処理モジュールは、画像出力装置に対応するデバイス識別情報に基づいて、前記データベースから前記依存処理モジュール識別情報と前記プロセス識別情報とを取得することを特徴とする請求項21乃至24のいずれかに記載の記憶媒体。

【請求項26】 前記デバイス識別情報を双方向インタフェースを介して前記画像出力装置から取得するデバイス識別情報取得モジュールを更に供給するデバイスドライバプログラムを格納することを特徴とする請求項25記載の記憶媒体。

【請求項27】 情報処理装置に新規に画像出力装置による出力を制御するためのデバイスドライバをインストールするインストールモジュールをデバイスドライバプログラムに更に含み、

前記インストールモジュールは、新規にインストールする画像出力装置のデバイス識別情報に対応する前記依存処理モジュールがすでにシステム内に組み込まれている場合は、前記依存処理モジュールを新たに組み込まないことを特徴とする請求項21乃至26のいずれかに記載の記憶媒体。

【請求項28】 前記インストールモジュールは、新規にインストールする画像出力装置のデバイス識別情報に対応する前記依存処理モジュールがすでに組み込まれている場合は、新規の画像出力装置に対応するアイコンを作成することを特徴とする請求項27記載の記憶媒体。

【請求項29】 前記インストールモジュールは、前記データベースのバージョン情報を判断し、新規にインストールしようとするバージョン情報の方が新しい場合は、データベースを上書きすることを特徴とする請求項27乃至28のいずれかに記載の記憶媒体。

【請求項30】 前記処理識別情報は、前記依存処理モジュールに対応するすべてのデバイス識別情報の中で最も古いデバイス識別情報と同じ値であることを特徴とする請求項25記載の記憶媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、画像出力装置に印刷データを送信する情報処理装置、情報処理方法、デバイスドライバプログラムが格納された記憶媒体に関するものであり、特に、画像情報を受け取るインクジェットプリンタや電子写真プリンタ等の画像出力装置に画像情報を出力するホストコンピュータ、その制御方法およびコンピュータ読み取り可能な印刷制御プログラムを格納した記憶媒体に関するものである。

【0002】

【従来の技術】一般に、画像情報を受け取る画像出力装置に適用される従来のプリンタドライバソフトウェア（デバイスドライバ）では、基本的に画像出力装置の解

像度／印字色などを除いては、共通のデータ書式に従った画像情報を生成して画像出力装置に送信し、個別の画像出力装置特有の実装に伴って必要となるデータ加工は個別の画像出力装置で行われるのが一般的であった。

【0003】ところが近年、インクジェットプリンタ等の画像出力装置が広汎に用いられるようになったことに伴い、これらの装置をより安価な価格で提供することが求められている。また一方で、これら画像出力装置に画像情報を送信するホストコンピュータの情報処理能力は急激に増大している。かかる背景に伴って、画像出力装置内部で行われていた処理の一部を、ホストコンピュータの動作により実現する画像出力装置用ドライバソフトウェアに移行せしめることにより、画像出力装置のハードウェア量を低減し、ひいては装置の価格を抑えることが試みられるようになってきている。

【0004】実際に、このような試みが特願平11-075465号に記載されている。この先行の出願の中では、画像出力装置等に異なる画像データ加工処理をドライバソフトウェアで行っていることが記載されている。また、そのときに画像出力装置毎の処理を、共通のインタフェースを持つモジュール群として作成し、画像出力装置の識別情報を参照し、その情報に応じてモジュール群を切り替える形で実装されている。

【0005】最近では、接続された出力機器に対応した処理毎に作成された上記モジュール群は、以下の手順で呼び出され、実行されている。

I. 画像情報を送信すべき画像出力装置の識別情報を、画像出力装置とホストコンピュータとの双方向通信を介して、あるいは画像を出力しようとする利用者（ユーザ）が設定することによって取得される。

II. この画像出力装置の識別情報を検索キーとして、コアダライブ中のデータベースを参照し、得られる依存モジュールの識別情報を取得する。

III. 依存モジュールの識別情報を基づいて、コアダライブは、画像出力装置に依存した処理を行う依存モジュールを選択する。

IV. 一般に、コアダライブと依存モジュールの開発は、別々に行われる。そのため、コアダライブにより呼び出される依存モジュールが、実際に画像情報を送信すべき画像出力装置に対応した処理が行われているかどうかの検査を依存モジュール側でも行う必要がある。そのため選択されたモジュール内部では、コアダライブから画像出力装置の識別情報を受け取りその検査を行う。

V. コアダライブから受け取った画像情報を画像出力装置に応じた形に加工し、出力する。

【0006】

【発明が解決しようとする課題】しかしながら、上述したシステムでは、従来の依存モジュールの処理で対応できる新規の画像出力装置を追加したい場合であっても、従来の依存モジュールで行われる出力装置の識別情報を

検査する工程で、画像出力装置の識別情報が異なるため、正常な動作が行われないという問題があった。そのために、従来の依存モジュールの内部を変更する必要がある。

【0007】本発明の目的は、上記従来の依存モジュールの処理で対応できる新規の画像出力装置を追加する場合に、依存モジュールの書き換え作業を行わずに、依存モジュール内部の検査工程を正常動作させることである。

【0008】

【課題を解決するための手段】本発明は、画像出力装置に送信すべき画像情報を生成する情報処理装置であって、入力される描画情報に対して、前記画像出力装置のタイプに共通の処理を行う共通処理手段と、前記画像出力装置のタイプに依存する処理を行い、前記画像情報を生成する複数の依存処理手段と、それぞれの画像出力装置に対して、使用する依存処理手段を識別する依存処理手段識別情報と前記依存処理手段に渡すべき処理識別情報とを保持するデータベース管理手段とを有し、前記共通処理手段は、前記依存処理手段識別情報に基づいて使用する依存処理手段を選択し、選択された依存処理手段に対して前記処理識別情報を渡すことにより、前記依存処理手段による内部検査を行わせるよう制御するものである。

【0009】また、前記データベース管理手段は、更に、前記共通処理手段で使用する色補正データを保持しているものである。

【0010】また、前記共通処理手段は、オペレーティングシステムから受け取るディスプレイ表色系データを、前記色補正データに基づいて、画像出力装置における表色系データに変換処理するものである。

【0011】また、前記共通処理手段は、オペレーティングシステムから受け取るRGBラスタデータを、前記色補正データに基づいて、CMYKもしくはCMYK淡C淡Mのラスタデータに変換処理するものである。

【0012】また、前記共通処理手段は、画像出力装置に対応するデバイス識別情報に基づいて、前記データベース管理手段から前記依存処理手段識別情報と前記処理識別情報とを取得するものである。

【0013】また、前記デバイス識別情報を双方向インタフェースを介して前記画像出力装置から取得するデバイス識別情報取得手段を更に有するものである。

【0014】また、情報処理装置に新規に画像出力装置による出力を制御するためのデバイスドライバをインストールするインストール手段を更に有し、前記インストール手段は、前記データベース管理手段により新規にインストールする画像出力装置のデバイス識別情報に対応する前記依存処理手段がすでに組み込まれている場合は、前記依存処理手段を新たに組み込まないものである。

【0015】また、前記インストール手段は、前記データベース管理手段により新規にインストールする画像出力装置のデバイス識別情報に対応する前記依存処理手段がすでに組み込まれている場合は、新規の画像出力装置に対応するアイコンを作成するものである。

【0016】また、前記インストール手段は、前記データベース管理手段のバージョン情報を判別し、新規にインストールしようとするバージョン情報の方が新しい場合は、データベース管理手段を上書きするものである。

【0017】また、前記処理識別情報は、前記依存処理手段に対応するすべてのデバイス識別情報の中で最も古いデバイス識別情報と同じ値である。

【0018】また、本発明は上記を実現するための方法または上記を実現するためのプログラムを格納した記憶媒体により実現する。

【0019】

【発明の実施の形態】本発明に係る実施の形態を説明する。図1は、本発明に係る画像出力装置と情報処理装置の構成例を表す図である。

【0020】図1において、100は情報処理装置本体でありホストコンピュータに相当する。200は周辺装置を含めたクライアント全体を示す。また、300は画像処理装置本体である。

【0021】情報処理装置100において、101はMPU（制御部）であり、DRAM103に格納されている制御手順に従って情報処理装置の全体制御を司る。102はシステム全体の格構成を接続するブリッジである。109はMPU101が実行するBIOSプログラムやフォントデータを格納するROMである。

【0022】103はMPU101が実行するプログラムやデータ等をROM109からロードし、1次記憶するDRAMであり、またMPU101のワークメモリとしても機能する。104はシステムバスとメモリバス、MPUを接続するブリッジ、105はCRTにグラフィック情報を表示するための制御機能を備えたグラフィックアダプタ（表示制御部）である。

【0023】106は外部記憶装置202とのインターフェースを司るHDDコントローラ、107はキーボードとのインターフェースとを司るキーボードコントローラ、108はIEEE1284規格に従って記録装置300との間の通信を司る、双方向通信可能なパラレルインターフェイスである通信I/F（入出力部）である。また、108はUSB（Universal Serial Bus）の規格に従った双方向通信可能なシリアルインターフェースであってもよい。

【0024】110は記憶媒体読み込み部であり、F、D、CD-ROM、CD-R、光ディスク、ZIP、DVD等の記憶媒体からプログラムやデータなどを読み込んでDRAMにロードし、必要に応じて、HDDなどの外部記憶装置202に記憶する。

【0025】111はネットワークや公衆回線との通信を行うNet通信部であり、ターミナルアダプタ（TA）やモデムに相当する。

【0026】ホスト装置100には、グラフィックアダプタ105を介して操作者にグラフィック情報等を表示する表示装置201（表示部）が接続されており、本実施の形態例では接続線駆動管（CRT）表示装置となっている。また表示装置はCRTではなく、液晶表示装置であるLCDであってもかまわない。更に、プログラムやデータが格納された大容量記憶装置であるハードディスクドライブ（HDD）装置202、キーボードやマウスなどの指示入力装置203（指示入力部）が接続されている。

【0027】画像出力装置本体300において、301はIEEE1284規格に従ってホスト装置100との通信を司る、双方向通信可能なパラレルインターフェースである通信I/Fである。また、300はUSB（Universal Serial Bus）の規格に従った双方向通信可能なシリアルインターフェースであってもよい。

【0028】302はMPUであり、画像出力装置全体を制御する制御部であり、また情報処理装置から受信した制御データを含む画像情報を解析し、印刷部306へのビデオ信号の出力制御を行う。303はMPU302が実行する制御プログラムやフォントデータ（アウラインフォントやビットマップフォント）を格納するROMである。305は、MPU302が制御するプログラムをROM303からロードし、一時記憶するRAMであり、MPU302のワークメモリとして機能する。304は、格構成要素間のデータ通信を行うデータバス、システムバスである。

【0029】印刷部306は、MPU302により送られるビデオ信号に基づいて記録媒体に記録画像を生成する。また、307は給紙部であり、記録媒体を給紙する機構である。308は排紙部であり、印刷部306で画像生成された記録媒体を排紙する機構である。

【0030】また、本実施例では、画像出力装置は、情報処理装置のドライバによりイメージ展開されたビットマップデータを簡単な制御データを付加した印刷データに基づく印刷を行うインクジェットプリンタとして説明しているが、これに限り必要はなく、情報処理装置からの印刷データを受け取り、画像出力装置内でイメージ展開して印刷を行う電子写真プリンタや、情報処理装置からの印刷データを図示省略した公衆回線を介してファクシミリ送信したり、ファクシミリ受信した印刷データを伸張して印刷を行うファクシミリ装置、更にはそれらの複数の機能を含む複合機能装置であってもよい。

【0031】次いで、以上の構成を備える本実施の形態における制御ソフトウェアの構成を具体的に図面を参照

しつづ説明する。まず、ホスト装置100に設置されるドライバソフトウェアを図2を参照して説明する。図2は本実施例の形態例のドライバのソフトウェアの構成を説明するための図である。

【0032】本実施例の形態例のドライバソフトウェアは、図2のCに示される階層であらわされる。

【0033】図2において、Aはアプリケーションソフトウェア（以下APPと称す）の階層、Bはオペレーションシステム（以下OSと称す）を構成する階層、Cはドライバソフトウェアの階層を示す。

【0034】記録装置にAPPのA1に示すプログラムにおいて、画像を出力しようとする場合、APPはOSの描画処理インターフェースB1に対して、文字・線分・図形・ビットマップなどの描画命令（API）を発行する。

【0035】画面・紙面を構成する描画命令が完結すると、OSは各描画命令（API）を、C11～C1nであらわされるドライバ内部の装置固有の描画手段を呼び出しつつ、OSの内部形式に変換する。この場合には、RGB各色を8ビット／画素で表現される点順次のラスターデータを描画情報としてC4の色補正／色変換手段に渡す。

【0036】色補正／色変換手段C4の処理を行う前に、データベース読み込み手段C3以後の処理で必要となるパラメータ群を保持したデータベースを、画像出力装置300のデバイスIDを検索キーとして、情報処理装置本体のDRAM103にロードする。

【0037】なお後述するように、この検索キーとなる画像出力装置のデバイスIDは、情報処理装置200と画像出力装置300とを結ぶ双方向インタフェースを介して得られる。また、画像を出力しようとする利用者が画像出力装置300を選択する際に、デバイスIDを選択するようにしてもよい。

【0038】外部記憶装置（HDD）からDRAMにロードされたデータベース405の内容を図3に示す。図3において、400はデバイスIDであり、取得したデバイスIDと同様のデバイスIDを有するデータセットが後述するように検索される。401はOSから受け取る描画情報を画像出力装置に送信すべき画像情報として色補正／色変換処理を行う際に用いられるデータである。402は、中間表現が必要な画像情報を生成する際に用いられるハーフトーニングデータである。403は図2の仮想スイッチC61、C62で画像出力装置の固有の特性に依存する依存モジュールを選択するために必要な、依存モジュールを識別するデータである。依存モジュールは、画像出力装置に渡される画像データ加工処理を行うモジュールである。モジュール識別データは、依存モジュールを識別するIDであってもよいし、依存モジュールのファイル名であってもよい。404は、選択された依存モジュールに渡すプロセスIDである。プ

ロセスIDについては後述する。

【0039】本発明のデータベース405は、1つのデバイスIDに対して、このように色補正／色変換処理用データ、ハーフトーニングデータ、モジュール識別データ、プロセスIDデータを対応させており、本データベースは、このようなデータセットを複数組有している。このデータベースは、各モジュールと同様に、Windows OS（米国マイクロソフト社の登録商標）では、Windows フォルダ内のSystemフォルダ内に格納されている。なお、データベースには、現在のバージョンで対応可能なすべてのプリンタに対するデバイスIDのデータセットが管理されており、新規にプリンタが追加されるとき、つまり新規プリンタ用のドライバがインストールされるときには、このデータベース内を検索して新規プリンタに対応するデバイスIDに基づいて必要な各モジュールを判断し、現在インストールされているモジュールの中で足りないモジュールのみを、ドライバインストールが保持されているフォルダからWindows フォルダのSystem フォルダに対してインストールすることになる。このときデータベース自体は更新されることはない。ただし、ドライバのバージョン自体がバージョンアップされる場合、つまり、現在のデータベースのバージョン情報よりも新しいバージョン情報をもったデータベースのインストールの場合は、ドライバのインストールを行う際に、データベース自体を新規なバージョンのデータベースで書き換えることにより置き換える。

【0040】このように本発明のデータベースは構成されている。なお、従来のデータベースにはプロセスIDを持っていないため、後述する依存モジュールでの検査処理でエラーが発生するか、依存モジュールを新規に書き換える必要があった。

【0041】本実施例では、モジュール識別データ403とプロセスIDデータ404は動的にロードされるが、このデータ群をドライバのコード中に直接ハードコーディングしてもよい。

【0042】図2に示すC4の色補正／色変換手段では、データベース中のデバイスID400に対応する色補正処理・色変換処理用データ401を参照して、デバイスの色特性を補正／ドライバ内部の表色系からデバイス固有の表色系への変換を行う。表色系はデバイス固有であり、電子写真プリンタや中級インクジェットプリンタはCMYK各色8ビットとなり、高級インクジェットプリンタはCMYKに淡C（淡いシアン）、淡M（淡いマゼンタ）が追加され6色となる。よって、この色補正／色変換手段C4は、CMYK 4色もしくはCMYK淡C淡M 6色の各色8ビット／画素で表現される点順次のラスターデータとしてハーフトーニング手段C5に渡す。

【0043】ハーフトーニング手段C5では、データベース中のハーフトーニングデータ402を参照して、デバイスの各画素の状態を表す量子化量への変換を行い、

各色1ビットあるいは4ビット／ピクセルの線順次データとして、スイッチC61、C62で選択されたモジュールへ渡す。

【0044】ここで、色補正／色変換手段およびハーフトーニング手段は、デバイスドライバの中心部分であり、処理自体はいずれのデバイスに対しても行われるものであるため、コアダライバと呼ばれる。コアダライバは、データベースからデバイスIDに依りて取得されるパラメータ（色補正・色変換処理用データ及びハーフトーニングデータ）に基づいて、色変換データベース（図示省略）やハーフトーニングデータベース（図示省略）を参照して色補正／色変換処理、ハーフトーニング処理を行うことになる。色変換データベース及びハーフトーニングデータベースは、デバイス固有であり、1つのデバイスに対して1つのデータベースが用意されている。

【0045】モジュールC71、C72、C73の部分には画像出力装置300に依存した画像情報加工作業が多いことから、この部分に統一した入出力インターフェースをもつ構造としている。こうすることで、情報供給装置100の個別の実装に依存するプログラム部品の作成を容易なものとし、かつドライバソフトウェアの異機処理部分を個別の情報供給装置100から独立した構造にすることが可能となる。

【0046】本実施例では、色補正／色変換手段（色補正／色変換モジュール）とハーフトーニング手段（ハーフトーニングモジュール）をコアダライバと定義し、デバイスに依存するモジュールを依存モジュールと定義し、プリントドライバ自体は、コアダライバと依存モジュールとから構成されている。

【0047】コアダライバは、仮想スイッチC61、C62において、C71、C72、C73のいずれかの依存モジュールを選択する。その選択方法を図6で具体的に説明する。

【0048】ステップS100では、コアダライバは画像情報を送信すべき画像出力装置300から双方向インターフェースを介して、もしくはユーザにより指定された画像出力装置に対応するデバイスIDを取得し、このデバイスIDを検索キーとして、データベース405を検索する。対応するデバイスID400が検索されると、そのデータセット内に保持されている、画像出力装置のモジュール識別データとそのプロセスIDを取得し、コアダライバで保持する。

【0049】ステップS101では、コアダライバは、取得したモジュール識別データを用いてスイッチC61、C62に、依存モジュールを選択する。

【0050】ステップS102では、コアダライバは、選択した依存モジュールに対してステップS100で取得したプロセスIDを渡す。その後、依存モジュールでプロセスIDの検査が行われ、正常である場合は、続いてコアダライバは生成した線順次のラスタデータを依存

モジュールに渡す。

【0051】ここで、本発明で導入した、プロセスIDについて具体的に説明する。

【0052】図4は、接続される画像出力装置とその処理を行うモジュール群の対応関係の例を表している。ここでいうモジュールとは依存モジュールのことである。

【0053】モジュールAは、画像出力装置A310（デバイスID：100）での依存処理に対応している。モジュールBは、画像出力装置B311（デバイスID：101）、画像出力装置C312（デバイスID：102）での依存処理に対応しており、モジュールCは、画像出力装置D313（デバイスID：103）での依存処理にそれぞれ対応している。

【0054】このように、画像出力装置はいろいろな製品スペックがあり、紙送り量やヘッド位置の調節など、デバイスに応じて異なっているために、デバイスに依存したモジュールがそれぞれ対応しているが、図4に示すように同系統の画像出力装置（例えば、画像出力装置Bと画像出力装置C）では同様の依存モジュールが対応することができる。

【0055】本実施例では、プロセスIDとは、依存モジュールに対応する画像出力装置が複数存在したときの一番古い画像出力装置のデバイスIDを用いている。また、一番古い画像出力装置のデバイスIDを用いずに、デバイスIDではないユニークなプロセスIDをもたせてもよい。いずれにしても、同じ依存モジュールを使用する画像出力装置では、同じプロセスIDを用いることになる。

【0056】一番古い画像出力装置のデバイスIDを用いる場合は、図4の例では、モジュールBに渡すプロセスIDは、画像出力装置B311のデバイスID（101）となる。モジュールAに関しては対応している画像出力装置は1つみである。この時、プロセスIDは、画像出力装置AのデバイスID（100）となり、モジュールCに関しては、画像出力装置DのデバイスID（103）となる。

【0057】スイッチC61、C62で選択されたC71、C72、C73のいずれかの依存モジュールは、画像を出力しようとする利用者に選択された画像出力装置300の種類に応じた、データ圧縮／コマンド付加処理を行い、システムのプリントサーバB2に渡す。依存モジュールの具体的な処理を図7を用いて説明する。

【0058】ステップS200では、依存モジュールは、依存モジュール自身が画像出力装置に対応した処理に対応しているかの検査を行うために、ドライバから渡されたプロセスIDのエラーチェックをおこなう。これは依存モジュールはプロセスIDが書き込まれてコンパイルされており、このプロセスIDとコアダライバから渡されたプロセスIDが一致しているかの判断を行うことにより実現する。エラーチェックにより正常と判断さ

れた場合は、ステップS201に処理を進める。一方、エラーと判断された場合は、ステップS203に処理を進め、ステップS203において、エラーメッセージを表示部に表示させて終了する。なおこの場合、コドライバはラスタデータを依存モジュールには渡さずに消去する。

【0059】ステップS201では、依存モジュールは、コドライバによりハードウェア処理されたラスタデータを、データ転送速度を高めるためにデバイスに依存した圧縮形式で圧縮を施す。圧縮形式は画像出力装置に応じて複数あり、インクジェットプリンタ系は主にパックビット(Pack Bits)形式による圧縮を施し、電子写真プリンタ(LBP)系はパックビット形式やその他の圧縮形式をデータに応じて施す。

【0060】ステップS202では、依存モジュールは、画像出力装置固有の制御コマンドをステップS201で生成されたデータに付加する。この制御コマンドとは、例えば、インクジェットプリンタでは、紙送り量やヘッド位置の調節のための制御コマンドであり、電子写真プリンタでは、プリンタ名、解像度情報、階調情報、フント名などの情報である。このようにして、依存モジュールによりラスタデータに圧縮処理や制御コマンドの付加が行われて画像出力装置に送信すべき画像情報が生成される。

【0061】図2のシステムのプリンタスアーバB2は、通信I/F108を介してIEEE1284に定められた手順に従って、画像情報を画像出力装置300へ転送を行う。

【0062】従来、一般的にプリンタドライバの開発において、コドライバと依存モジュールの開発は、別々に行われ、依存モジュールを呼び出す側(コドライバ側)は共通のインタフェースを用いるために、依存モジュールが正しいかの検査を意識せずに使用できる必要性がある。このため、この検査を依存モジュール内部で行う必要があった。すなわち、依存モジュールで同じ処理を行う新規画像出力装置が開発時に追加される場合では、従来は、コドライバから渡されるデバイスIDを用いたエラーチェック(検査)を依存モジュールの中で行う部分を新規に作成する必要があった。

【0063】しかし、本実施例で上述したように、同じ依存モジュールを使用する場合はデバイスIDではなく、同じプロセスIDを受け渡しに用いることにより、既存のモジュールと同様の処理を行う新規機種を追加する場合には、依存モジュール自体は変更する必要がなくなり、デバイスIDからプロセスIDを取得するためのデータベースを適に作成すれば、このプロセスIDをモジュールに渡すことでエラーチェック部分を書き換える必要がなくなり、開発工程が大幅に減少することが可能となる。また、プロセスIDとして、本実施例のように、同様の依存モジュール処理を行う一昔古い画像出力

装置のデバイスIDを用い、すでに存在する従来の依存モジュール自体も変更することなく、データベースのみを適に作成すればよくなる。

【0064】図5は、新規に画像出力装置E(デバイスID:104)が、本システムに追加された場合、つまり本システムに新規に画像出力装置Eのドライバ(依存モジュール部分)がインストールされる場合の画像出力装置とその処理を行う依存モジュール群の対応関係の例を表している。

【0065】この時、本発明のシステムにおいては、画像出力装置EのデバイスID:104が与えられたときにプリンタドライバに渡すプロセスIDを、対応モジュールBのプロセスID101を返すようにデータベース400内のプロセスIDデータ404を作成し、かつ、モジュール識別データ403をモジュールBの識別データを返すように開発時にデータベースを作成する。つまり、開発工程で新規の画像出力装置を追加する場合は、新規の画像出力装置のデバイスIDに対応する依存モジュールに対応するプロセスIDを、データベースの本デバイスIDのデータセットに追加し、あとは従来通り色補正・色処理用データ、ハードウェア用データ、モジュール識別データを含むデータセットのデータベースを作成する。開発時にはこのような開発工程が行われる。

【0066】実際にドライバをインストールからドライバを情報処理装置にインストールする処理について説明する。

【0067】まず、情報処理装置内にドライバがまったくインストールされていない場合、つまりはじめてドライバのインストールを行う場合には、コドライバ、インストールする画像出力装置の依存モジュール、インストーラのバージョンのデータベース、色補正/変換処理用データベース(図示省略)がそれぞれWindowsフォルダのSystemフォルダにインストールされ、新規プリンタアイコンの作成が行われる。

【0068】次に、別の画像出力装置の追加がユーザにより指示された場合には、指示された画像出力装置のデバイスIDを用いてインストールされているデータベースを検索してモジュール識別データ403を取得し、そのモジュール識別データ403(本実施例では、依存モジュールのファイル名)で識別される依存モジュールがSystemフォルダにインストールされているかを判断し、インストールされていない場合は、依存モジュールをインストールして、プリンタアイコンの作成を行う。このように、2回目以降はコドライバとデータベースのインストールは行わずに依存モジュールのインストールを行えばよい。

【0069】また、更に画像出力装置の追加がユーザにより指示された場合には、同様に指示された画像出力装置のデバイスIDを用いてデータベースを検索し、依存モジュールがSystemフォルダにインストールされている



かを判断する。ここで同じタイプの画像出力装置の場合には、依存モジュールは同様のものを使用できることが多いため、すでにインストールされていることが考えられる。この場合は、依存モジュールのインストールは行わずに、画像出力装置を示すプリンタアイコンの生成と、そのデフォルト設定を行って終了する。つまり、本発明では、すでに同じプロセスIDで使用する依存モジュールが付いてインストールされている場合は、新たなモジュールの追加は行う必要がなく、プリンタアイコンの作成処理だけで済み、開発工程だけでなく、情報処理装置のメモリ効率がよくなるという効果も得られる。

【0070】ここで上記のインストールを行う際には、必ず現データベースのバージョン情報と、インストールしようとするドライバ（実際にはドライバインストールを示すデータベース）のバージョン情報との比較が行われ、現データベースのバージョン情報が同じか大きい（新しい）場合に、上記の処理が行われる。現データベースのバージョン情報よりもインストールしようとするドライバのバージョン情報が新しい場合は、データベースの上書きが行われる。ただし、それまでに情報処理装置内にインストールされているドライバ（コアダライバ、依存モジュール）はそのまま保存される。なぜならバージョン情報の新しいデータベースないには、今までサポートしてきたドライバに対応するデバイスIDのデータセットが管理されているため、問題なく使用を続けることができるからである。

【0071】また、従来のコアダライバでは、デバイスIDを用いてデータベースを検索し、デバイスIDに対応するモジュール識別データのみを取得し、モジュール識別データに基づいて依存モジュールを選択し、選択された依存モジュールに対して、デバイスIDを渡していた。そして依存モジュールは受け取ったデバイスIDが自身の内部に記載されたコンパイルされているIDと同じであるかのエラーチェックを行っている。

【0072】これに対して、本発明のコアダライバは、デバイスIDを用いてデータベースを検索し、デバイスIDに対応するモジュール識別データとプロセスIDとを取得し、モジュール識別データに基づいて依存モジュールを選択し、選択された依存モジュールに対して、プロセスIDを渡すことになる。そして依存モジュールは受け取ったプロセスIDが自身の内部に記載されたコンパイルされているIDと同じであるかのエラーチェックを行う。

【0073】このように、コアダライバは、図6のステップS102で選択された依存モジュールBにプロセスIDを渡すことで、図7のステップS200の入力プロセスIDの検査部分を書き換える必要がなくなり、開発工程が少なくなるという効果が得られる。

【0074】また、本発明は、前述したコアダライバ、依存モジュール及びデータベースを含むプリンタドライ

バのプログラムコードを記憶した記録媒体を、システムあるいは装置のコンピュータ（またはCPUやMPU）が記録媒体に格納されたプログラムコードを記憶媒体読み込み部110を用いてDRAM103に読み出し、MPU101が実行することによっても達成される。図8にその一例を示す。

【0075】800は、本発明の処理をプログラムコード化した記憶媒体である。801は、インストーラとして機能するインストールモジュールである。このプログラムを実行することにより、インストール処理が行われる。802は、コアダライバモジュールであり、色補正／色変換処理モジュールC4とハーフトーニングモジュールC5を含むものである。803は依存モジュールであり、デバイスに依存する複数の依存モジュールが格納されている。804は、図3に示すデータベースであり、またバージョン情報を有している。805は取得モジュールであり、双方向インタフェースを介して画像出力装置からデバイスIDを取得するためのモジュールである。806は色変換データベースであり、色補正／色変換処理を実行する際に参照されるデータベースである。本それぞれのモジュールを供給する記憶媒体はこのように構成されている。

【0076】この場合、記憶媒体から読み出されたプログラムコード自体が前述した実施形態の機能を実現することになり、そのプログラムコードを記憶した記録媒体は本発明を構成することになる。

【0077】プログラムコードを供給するための記憶媒体としては、例えば、フロッピー（登録商標）ディスク、ハードディスク、光ディスク、光磁気ディスク、CD-ROM、CD-R、CD-RW、磁気テープ、不揮発性のメモリカード、MO、DVD、ROMなどを用いることができる。

【0078】また、コンピュータが読み出したプログラムコードを実行することにより、前述した実施形態の機能が実現されるだけでなく、そのプログラムコードの指示に基づき、コンピュータ上で稼動しているOS（オペレーティングシステム）などが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれる。

【0079】さらに、記憶媒体から読み出されたプログラムコードが、コンピュータに挿入された機能拡張ボードやコンピュータに接続された機能拡張ユニットに備わるメモリに書き込まれた後、そのプログラムコードの指示に基づき、その機能拡張ボードや機能拡張ユニットに備わるCPUなどが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれる。

【0080】また、本発明の機能処理を本情報処理装置であるコンピュータで実現するために、該コンピュータにインストールされるプログラムコード自体も本発明を

実現するものである。つまり、本発明は、本発明の機能処理を実現するためのコンピュータプログラム自体も含まれる。

【0081】このコンピュータプログラムの供給方法としては、上述したように、FDやCD-ROMに記憶させて、コンピュータに読み取らせて該コンピュータ内にインストールされる場合に限らず、クライアントコンピュータのブラウザ（図示省略）によりNet通信部111を介してインターネットのホームページに接続し、該ホームページから本発明のコンピュータプログラムそのもの、もしくは圧縮され自動インストール機能を含むファイルをダウンロードすることによっても供給できる。また、本発明のプログラムを構成するプログラムコードを複数のファイルに分割し、それぞれのファイルを異なるホームページからダウンロードすることによっても実現可能である。つまり、本発明の機能処理をコンピュータで実現するためのプログラムファイルを複数のユーザに対してダウンロードさせるWWWサーバも、本発明に含まれるものである。

【0082】また、本発明のプログラムを暗号化してFD等の記憶媒体に格納してユーザに配布し、所定の条件をクリアしたユーザに対し、インターネットを介してホームページから暗号化を解く鍵情報をダウンロードさせ、その鍵情報を使用することにより暗号化されたプログラムを実行してコンピュータにインストールさせて実現することも可能である。

【0083】以上説明したように、本実施例によれば、プロセスIDを用いて依存モジュールに内部検査（エラーチェック）を行わせることにより、従来の依存モジュールの処理で対応できる新規の画像出力装置が追加されたときに発生する、従来の依存モジュールの書き換え作業をなくすことができる。

【0084】【発明の効果】以上説明したように、本発明によれば、処理識別情報を用いて依存処理手段に内部検査を行わせることにより、依存処理手段を改変することなく、複数の画像出力装置に対応させることが可能となる。

【図面の簡単な説明】

【図1】本発明に係るシステムブロック図である。

【図2】本発明の実施形態に係る情報処理システムで用いられるソフトウェアの構造を示すブロック図である。

【図3】本発明のドライバソフトウェアが保持するデータベースのブロック図である。

【図4】本発明に係るモジュールと画像出力装置の対応関係を示す図である。

【図5】本発明に係る新規の画像出力装置が追加されたときのモジュールと画像出力装置との対応関係を示す図である。

【図6】本発明に係るドライバソフトウェアがモジュール

ルを呼び出すための処理のフローチャート図である。

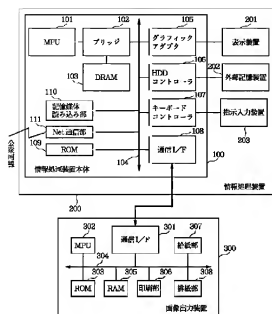
【図7】本発明に係るモジュールの処理のフローチャート図である。

【図8】本発明に係る各モジュールを供給するための記憶媒体のメモリブロック図である。

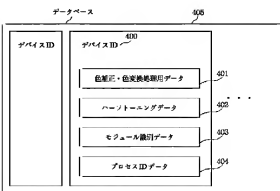
【符号の説明】

100 情報処理装置本体  
101 MPU  
102 ブリッジ  
103 DRAM  
104 バス  
105 グラフィックスアダプタ  
106 HDDコントローラ  
107 キーボードコントローラ  
108 通信I/F  
109 ROM  
110 記憶媒体読み込み部（FDD）  
111 Net通信部（TA、モデム）  
200 情報処理装置  
201 表示装置  
202 外部記憶装置（HDD）  
203 指示入力装置（キーボード）  
300 画像出力装置  
301 通信I/F  
302 MPU  
303 ROM  
304 バス  
305 RAM  
306 印刷部  
307 給紙部  
308 排紙部  
310 画像出力装置A  
311 画像出力装置B  
312 画像出力装置C  
313 画像出力装置D  
314 画像出力装置E  
400 デバイスID  
401 色補正・色変換処理用データ  
402 ハフマンデータ  
403 モジュール識別データ  
404 プロセスID  
405 データベース  
800 記憶媒体  
801 インストールモジュール  
802 コアドライバモジュール  
803 依存モジュール  
804 データベース  
805 取得モジュール  
806 色変換データベース

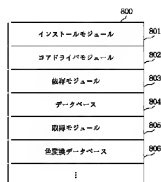
【図1】



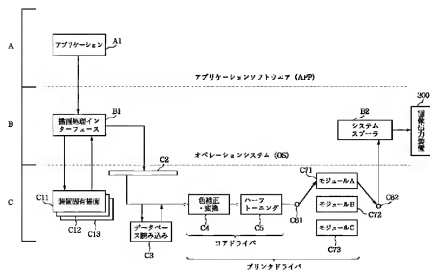
【図3】



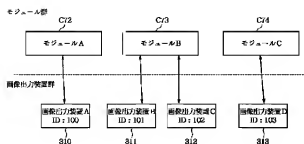
【図8】



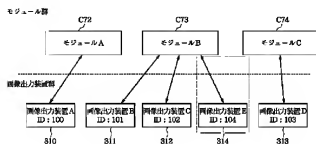
【図2】



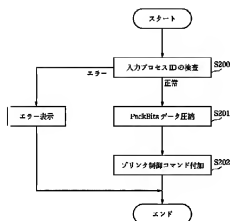
【図4】



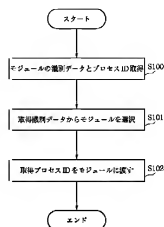
【図5】



【図7】



【図6】



フロントページの続き

Fターム(参考) 5B014 FA11  
 5B021 AA01 CC05 LL05  
 5B076 AA02 AC07 BB04 BB06  
 5C077 LL19 MP08 PP09 PP33 PP37  
 PQ22 SS02 SS06 TT02  
 5C079 BB03 LB02 MA01 NA03 NA17  
 PA03 PA05